

## Bagian 1

### Grammar CFG yang Dibangun

1.  $S \rightarrow NP VP$
2.  $NP \rightarrow Det N \mid RPR$
3.  $VP \rightarrow V \mid V NP \mid V PP \mid VP PP \mid V N \mid N PP \mid V PNP$
4.  $PNP \rightarrow N PP$
5.  $PP \rightarrow IN N$
6.  $V \rightarrow \text{"memberi"} \mid \text{"melihat"} \mid \text{"memfoto"}$
7.  $RPR \rightarrow \text{"Saya"}$
8.  $Det \rightarrow \text{"para"} \mid \text{"sang"} \mid \text{"si"}$
9.  $N \rightarrow \text{"tetangga"} \mid \text{"tahu"} \mid \text{"beruang"} \mid \text{"kacamata"} \mid \text{"piring"} \mid \text{"Ilham"}$
10.  $IN \rightarrow \text{"ke"} \mid \text{"dengan"} \mid \text{"dan"}$

```
13 print(grammarCFG.is_flexible_chomsky_normal_form())  
True
```

Didalam *grammar* tersebut ditambahkan 1 tag PNP untuk membuat *grammar* CFG berbentuk CNF. PNP sendiri terdiri dari N dilanjutkan oleh kata PP. Salah satu contoh pembangunan dari tag PNP adalah “beruang dengan kacamata”. PNP sendiri dibangun dikarenakan untuk membantu kalimat dengan keterangan tambahan yang berkaitan antara 2 objek seperti yang telah dicontohkan.

### Kalimat yang Akan Dicek

1. kalimatTest = 'Saya melihat beruang dengan kacamata'
2. kalimatTest = 'Saya memberi tahu dengan piring ke tetangga'
3. kalimatTest = 'Saya memfoto beruang bersama Ilham'

### Hasil Top Down untuk Kalimat 2

1. (S
2. (NP (RPR Saya))
3. (VP
4. (VP (V memberi) (PNP (N tahu) (PP (IN dengan) (N piring))))
5. (PP (IN ke) (N tetangga)))
6. (S
7. (NP (RPR Saya))
8. (VP
9. (VP (VP (V memberi) (N tahu)) (PP (IN dengan) (N piring)))
10. (PP (IN ke) (N tetangga)))

### Hasil Bottom up untuk Kalimat 2

1. (S

2. (NP (RPR Saya))
3. (VP
4. (VP (VP (V memberi) (N tahu)) (PP (IN dengan) (N piring)))
5. (PP (IN ke) (N tetangga)))
6. (S
7. (NP (RPR Saya))
8. (VP
9. (VP (V memberi) (PNP (N tahu) (PP (IN dengan) (N piring))))
10. (PP (IN ke) (N tetangga)))

## Analisa

Hasil dari percobaan menggunakan kalimat 2 terdapat ambiguitas dapat dilihat dari hasil *bottom up* maupun top down menghasilkan 2 pohon yang membuktikan kalimat tersebut memiliki ambiguitas. Ambiguitas dalam kalimat tersebut adalah pada bagian “tahu dengan piring”. Kedua metode menghasilkan bahwa “tahu dengan piring” memiliki arti tahu tersebut diantarkan dengan piring atau tahu dan piring diberikan kepada tetangga.

## Bagian 2

### Kalimat yang Digunakan

1. kalimatTest = 'monyet mengganggu warga Delhi'
2. kalimatTest = 'pemerintah amankan monyet berekor panjang'

### Analisa Kalimat 1

Pada kalimat 1 yang merupakan potongan kalimat dari *treebank* hasil dari parsing *bottom up*nya memiliki banyak ambiguitas, ambiguitas ini dikarenakan adanya suatu aturan grammar dimana kata NP bisa dilanjutkan oleh NP lagi. Sehingga ketika menggunakan metode *bottom up* hasilnya memiliki banyak perulangan pada aturan NP diikuti oleh NP. Sedangkan ketika menggunakan *viterbi parser* hasilnya sama seperti yang diharapkan oleh dataset.

### Analisa Kalimat 2

Hasil dari kalimat 2 juga tidak memiliki kesalahan yang sama pada kalimat 1 masih terpacu pada ambiguitas NP diikuti oleh NP juga. Hal ini dikarenakan grammar yang terbentuk merupakan hasil dari kalimat uji. Dan hasil dari *viterbi parser*nya sangat memuaskan karena dapat menghasilkan sebuah aturan yang diharapkan. Yaitu subjek diikuti oleh verb kemudian ada subjek dengan keterangan tambahan yaitu “monyet berekor panjang”.